CR-133987

[redacted]

(NASA-CR-133987) SPLINE SMOOTHING OF HISTOGRAMS BY LINEAR PROGRAMMING (Rice Univ.) 30 p HC $3.50 CSCL 12A

N73-31551

Unclas

G3/19 13557

ICSA

INSTITUTE FOR COMPUTER SERVICES AND APPLICATIONS

RICE UNIVERSITY

SPLINE SMOOTHING OF HISTOGRAMS BY LINEAR PROGRAMMING

BY

JOHN O. BENNETT

MATHEMATICAL SCIENCES DEPARTMENT

RICE UNIVERSITY

ABSTRACT

An algorithm for an approximating function to the frequency distribution is obtained from a sample of size n. To obtain the approximating function a histrogram is made from the data. Next ℓ_∞ and ℓ_1 Euclidean space approximations to the graph of the histogram using central B-splines as basis elements are obtained by linear programming. The approximating function has area one and is non-negative.

Institute for Computer Services & Applications

Rice University

Houston, Texas 77001

September 1972

Research supported under NASA Contract # NAS 9-12776

Index

2. Statement of Problem

Given a random variable x on a probability space P, let f(x) be the density function associated with x. Let

$$F(x) = \int_{-\infty}^{x} f(s)ds \tag{1}$$

be the cumulative density function associated with x. The problem is:

Given a random sample of size n, $\{x_1, \ldots, x_n\}$ can the density function f(x) be approximated by a smooth function using this data?

The first work done on this problem is by Benova, Kendall, and Stevetanov [4] in a function space setting. They define an approximation to f(x) called a Histospline by a homeomorphism of the ℓ_2 Hilbert Space of all histograms to a subspace of a Hilbert Space of smooth functions. In a recent paper by I.J. Schoenberg [14] he reconstructs the Histospline in a simpler setting and forms his splinogram with the variation-diminishing property using B-splines.

In this paper another approach is taken to approximate a histogram.

3. Splines

A spline is a mechanical device used by draftsmen to draw smooth curves. It consists of a piece of wood or plastic with lead weights placed on the points where it is desired that the spline pass through. These points are called knots. The differential equations for a bending beam with weights was solved by Holladay in [11]. This was one of the most important papers in the development of spline functions. The solution was piecewise cubic polynomials which had the first and second derivatives equal at each knot.

Actuarians have been using spline functions since the 1930's for smoothing life expectancy tables (see Greville [9] for a survey of the early work). Also, in the ship building industry they have been using these in moving weights around on beams (called lofting) to get the hull of a ship to match the design (see Berger et. al. [2]). The work that is the basis for most mathematical investigations of splines is I.J. Schoenberg's work [13]. For B-splines Greville [10] is the best reference.

A spline of degree r with m knots

$$x_1 \le x_2 \le \ldots \le x_m$$

is a function s(x)

(1) $s(x)$: $\mathbb{R} \to \mathbb{R}$, where $\{\text{real numbers}\} = \mathbb{R}$

(2) $s(x)|_{(x_j,\, x_{j+1})} = P_{rj}$, a polynomial of degree r, $j = 1, 2, \ldots, m$

(3) $s(x) \in C_{\mathbb{R}}^{r-1} = \{f \in \mathbb{R} \,|\, f^{(r-1)} \text{ is continuous}\}$.

Note that a spline of degree zero is a step function and a spline of degree one is a polygon. The advantage of using splines rather than polynomials to fit n data points

is a polynomial of degree n-1 or less is required while a spline of degree r, with r fixed, can be used and $r \ll n$. A very simple type of spline is the truncated power function;

$$x_+^m = \begin{cases} x^m, & \text{if } x > 0 \\ 0, & \text{if } x \le 0 \end{cases} .$$

Let $S_r(x_1, \ldots, x_m)$ be the set of all splines of degree r with m knotes. An important theorem in the theory of splines is

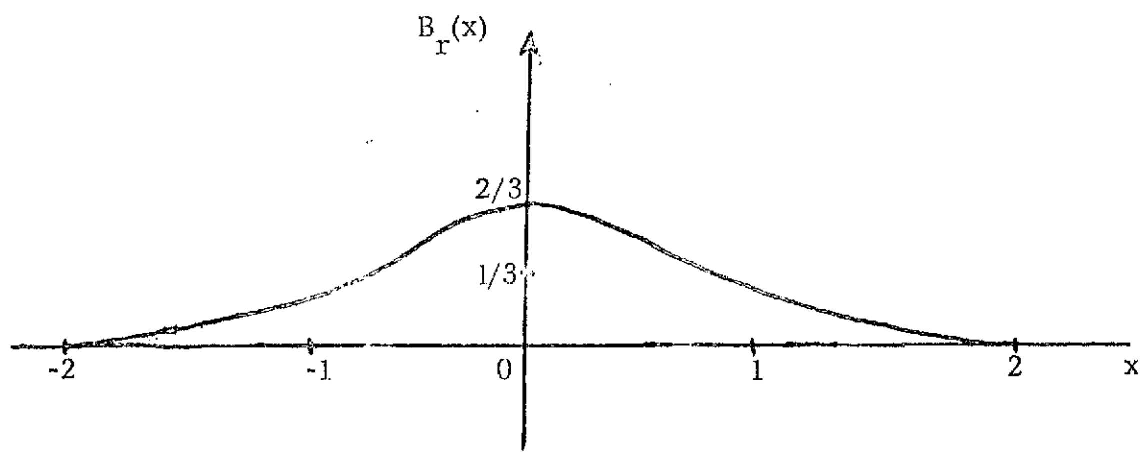
$s(x) \in S_r(x_1, \ldots, x_m) \; \exists ! \; P_r(x)$ (and) c_j such that

$$s(x) = P_r(x) + \sum_{j=1}^{m} c_j(x_j - x)_+^r , \quad \text{where } P_r(x) \text{ is a polynomial of degree } r.$$

See Greville [9]. But this form gives rise to ill-conditioned matrices when one actually solves for the c_j and $P_r(x)$ (see Schumaker[15]). We shall use the B-splines of Curry and Schoenberg [6]. For equally spaced knots of odd degree $r = 2k-1$, where k is a natural number, they are given by

$$\beta_r(x) = \frac{1}{r!} \sum_{j=-k}^{k} (-1)^{j+k} \binom{2k}{j+k} (j-x)_+^r \qquad (3)$$

using the form of Rosen [12]. For $k = 2$ i.e. for cubic B-splines the graph of $\beta_r(x)$ is



For this paper, we shall have $k = 2$. The function $\beta_r(x)$ is symmetric about $x = 0$, bell-shaped and non-negative on the interval $[-k, k]$. It is identically zero off its support $[-k, k]$. The properties of $\beta_r(x)$ are

$$\text{(a)} \qquad \beta_r(x) > 0 \quad , \quad 0 < |x| < k \quad , \tag{4}$$

$$\text{(b)} \qquad \beta_r(x) = \beta_r(-x) \quad , \tag{5}$$

$$\text{(c)} \qquad \beta_r(0) > \beta_r(x) \text{ for } x \neq 0 \quad , \tag{6}$$

$$\text{(d)} \qquad \sum_{i=-\infty}^{\infty} |\beta_r(i)| = \sum_{i=1-k}^{k-1} \beta_r(i) = \int_{-\infty}^{\infty} \beta_r(x)dx = 1 \tag{7}$$

It is obvious that these properties make the B-spline a natural candidate for a basis for a probability density function f(x) such that

$$\text{(a)} \qquad f(x) \geq 0 \quad , \tag{8}$$

$$\text{(b)} \qquad \int_{-\infty}^{\infty} f(x)\, dx = 1 \tag{9}$$

What is desired is a function $f_a(x)$ that satisfies (8) and (9) and is a good approximation to f(x). Let $f_a(x)$ be this approximating function

$$f_a(x) = \sum_{j=1}^{m} a_j \varphi_j(x)$$

where the φ_j are B-splines.

4. Algorithm for Histogram

The algorithm for the histogram goes as follows. The "n" observations are taken from a probability distribution and ordered such that

$$x_1 \le x_2 \ . \ . \ . \ \le x_{n-1} \le x_n \quad . \tag{10}$$

Note that if the sample is taken from a continuous distribution then restricted inequality may be placed between the data points. The next step is to determine where the knots are to be placed. In this paper they are placed at the integers between the data points and the first integer greater than x_n and the first integer less than x_1. Number these knots as follows

$$\bar{x}(1) < \bar{x}(2) \ . \ . \ . \ < \bar{x}(m) \quad . \tag{11}$$

Next construct the histogram for the n observations $\{x_i\}_{i=1}^{n}$ on the points

$$\tilde{x}_i = [\bar{x}(i) + \bar{x}(i+1)]/2 \quad , \ (i = 1,\ 2,\ \ldots,\ m-1)$$

$$\tilde{x}_0 = \tilde{x}_1 - 1 \quad , \quad \tilde{x}_m = \tilde{x}_{m-1} + 1 \quad .$$

5. Linear Programming

There is nothing novel about using linear programming for smoothing data. A method of finding the best line to fit data was used as early as 1820 by J.B. Fourier [8]. The simplex method of linear programming was introduced by G. Dantzig in the 40's [7]. Perhaps the first structuring of a similar problem for linear programming was by A. Charnes, W.W. Cooper et. al. [5]. The first use of linear programming for fitting data was by H.M. Wagner [17]. Much work has been done recently by Barrodale et. al. [1] in using linear programming for fitting data. For unconstrained approximation function this is the most efficient algorithm devised. The linear programming formulation of this paper is after J.B. Rosen [12]. The main general reference for linear programming in this paper is the book of A. Spivey and R.M. Thrall [16]. With the m points

$$(\bar{x}(i)\ ,\ y_i)\ {}_{i=1}^{m}$$

obtained from the Histogram Algorithm as input make the following definitions:

$$y^T = (y_1, \ldots, y_m)$$

and

$$\varphi_j(x) = \beta_r(x - \bar{x}(j))\ , \quad (j = 1, 2, \ldots, m)\ . \tag{14}$$

Let

$$a^T = (a_1, \ldots, a_m)$$

and

$$\varphi^T(x) = (\varphi_1(x)\,,\ \varphi_2(x)\,,\ \ldots,\ \varphi_m(x), 0\,,]$$

then the function that approximates these m data points is

$$f_a(x) = a^T\varphi(x) = \sum_{i=1}^{m} a_j\varphi_j(x) \ . \tag{15}$$

It is desired to have $f_a(x)$ approximate the data points in the ℓ_∞ and ℓ_1 norm where:

$$\|f_a(x) - y\|_\infty = \max_{i=1,2,\ldots,m} |f_a(x_i) - y_i| \tag{16}$$

$$\|f_a(x) - y\|_1 = \sum_{i=1}^{m} |f_a(x_i) - y_i| \ . \tag{17}$$

6. ℓ_∞ Norm

Now to formulate the ℓ_∞ norm as a linear programming problem let $\gamma \epsilon \mathbb{R}$ and (17) is equivalent to

$$\underset{a,\gamma}{\mathrm{Min}}\{\gamma \mid -\gamma \le f_a(x_i) - y_i \le \gamma, \quad i = 1, 2, \dots, m\} . \tag{18}$$

If γ^* is an optimal solution to (18) then

$$\gamma^* = \|f_a(x) - y\|_\infty ,$$

otherwise a smaller value could be found.

Now write (18) as

$$\left\{\begin{array}{l} \sum_{j=1}^{m} a_j\varphi_j(x_i) - y_i \ge -\gamma \\ \\ -\sum_{j=1}^{m} a_j\varphi_j(x_i) + y_i \ge -\gamma \end{array}\right. \qquad (i = 1, 2, \dots, m),$$

or

$$\left\{\begin{array}{l} \sum_{j=1}^{m} a_j\varphi_j(x_i) + \gamma \ge y_i \\ \\ -\sum_{j=1}^{m} a_j\varphi_j(x_i) + \gamma \ge -y_i \end{array}\right. \qquad (i = 1, 2, \dots, m). \tag{19}$$

Now (18) can be formulated as a linear programming problem as

$$\begin{array}{l} \mathrm{Min}\ \gamma \\ \ni \\ (19) \text{ holds} . \end{array} \tag{20}$$

To put (20) in a matrix form make the follow definations

$$F = \begin{bmatrix} \varphi_1(x_1) & \cdots & \varphi_1(x_m) \\ \cdot & & \\ \cdot & & \\ \cdot & & \\ \varphi_m(x_1) & \cdots & \varphi_m(x_m) \end{bmatrix}$$

and F is a m x m matrix.

Note that at this point we could solve the system of equations

$$Fa = y \tag{21}$$

to obtain the coefficients $\{a_i\}_{i=1}^{m}$ but this would give no assurance that equations (8) or (9) are satisfied by $f_a(x)$. Let

$$L^2_{[a,b]} = \{f \mid \left(\int_{-\infty}^{\infty} |f(x)|^2 dx\right)^{1/2} < \infty\} \ . \tag{22}$$

It is well known that

$$L^2_{[\bar{x}(1),\ \bar{x}(m)]}$$

is a Hilbert Space and thus has a well defined inner product. One could obtain the projection of the Histogram , which is in $L^2_{[\bar{x}(1),\ \bar{x}(m)]}$, on the $L^2_{[\bar{x}(1),\ \bar{x}(m)]}$ functions which integrate to one. This would satisfy equations (9) but not necessarily equation (8).

Continuing with the linear programming formulation let

$$C^T = (y^T, \; -y^T) \in \Re^{2m} \;,$$

$$W^T = (a^T, \; \gamma) \in \Re^{m+1} \;,$$

$$b^T = (0, \; \dots, \; 0, \; 1) \in \Re^{m+1} \;,$$

$$e^T = (1, \; \dots, \; 1) \in \Re^{m} \;.$$

Define A^T to be

$$A^T = \begin{bmatrix} F^T & e \\ \\ -F^T & e \end{bmatrix} \tag{23}$$

and A^T is a (2m) x (m+1) matrix. Requiring $\{a_i \geq 0\}_{i=1}^n$ (20) is

$$\begin{aligned} &\text{Min } b^T W \\ &\quad A^T W \geq C \\ &\qquad W \geq 0 \;. \end{aligned} \tag{24}$$

Since $W \geq 0$ the coefficients $\{a_i\}_{i=1}^m$ are found to be positive and so

$$f_a(x) = \sum_{j=1}^{m} a_j \varphi_j(x) \geq 0 \tag{25}$$

and equation (8) is satisfied. We now proceed to show equation (9) can be satisfied by adding one constraint. Because

$$\int_{-\infty}^{\infty} \varphi_j(x)dx = 1 \qquad (j = 3, \ldots, m-2)$$

$$\int_{-\infty}^{\infty} \varphi_2(x)dx = \int_{-\infty}^{\infty} \varphi_{m-1}(x)dx = .95833 \tag{26}$$

$$\int_{-\infty}^{\infty} \varphi_1(x)dx = \int_{-\infty}^{\infty} \varphi_m(x)dx = .5$$

it follows that

$$\int_{-\infty}^{\infty} f_a(x)dx = \int_{-\infty}^{\infty} \sum_{j=1}^{m} a_j\varphi_j(x)dx = \sum_{j=1}^{m} a_j \int_{-\infty}^{\infty} \varphi_j(x)dx =$$

$$.5a_1 + .95833a_2 + \sum_{j=3}^{m-2} a_i + .95833a_{m-1} + .5a_m \quad .$$

To satisfy (9) it is required that

$$\int_{-\infty}^{\infty} f_a(x)dx = 1$$

so set

$$d^T W = 1 \tag{27}$$

where

$$d^T = (0.5,\ .95833,\ 1,\ \ldots,\ 1,\ .95833,\ .5,\ 0) \in \mathbb{R}^{m+1} \ .$$

If equation (27) is satisfied then equation (9) is satisfied. The complete ℓ_∞ formulation of the problem is obtained by adding equation (27) to the constraints of equations (24).

The .95803 and .5 arrise from the fact that the two end basis functions have support outside of $[\bar{x}(1),\ \bar{x}(m)]$.

7. ℓ_1 Norm

For the ℓ_1 norm redefine γ

$$\gamma = \begin{bmatrix} \gamma_1 \\ \cdot \\ \cdot \\ \cdot \\ \gamma_m \end{bmatrix} .$$

Equation (17) becomes

$$\min_{a,\gamma} \left\{ \sum_{i=1}^{m} \gamma_i \,\middle|\, -\gamma_i \le f_a(x_i) - y_i \le \gamma_i , \quad i = 1, 2, \ldots, m \right\} . \tag{28}$$

If γ^* is the optimal solution to (28), then

$$\sum_{i=1}^{m} \gamma_i^* = \|f_a(x) - y\|_\infty .$$

But putting a subscript on the γ in equations (19) the ℓ_1 formulation of the constraints for equation (28) becomes

$$\begin{aligned} &\sum_{j=1}^{m} a_j\varphi_j(x_i) - \gamma_i \ge y_i \\ &\qquad\qquad\qquad\qquad i = 1, 2, \ldots, m \\ -&\sum_{j=1}^{m} a_j\varphi_j(x_i) - \gamma_i \ge -y_i \quad . \end{aligned} \tag{29}$$

The complete ℓ_1 formulation as a linear programming problem is

$$\operatorname{Min} \sum_{i=1}^{m} \gamma_i \tag{30}$$

$\ni$

(29) holds .

By defining

$$b^T = (0, \ldots, 0, e) \in \mathbb{R}^{2m}$$

and

$$A^T = \begin{bmatrix} F^T & I_n \\ & \\ -F^T & I_n \end{bmatrix}$$

and

$$W^T = (a^T, \gamma^T)$$

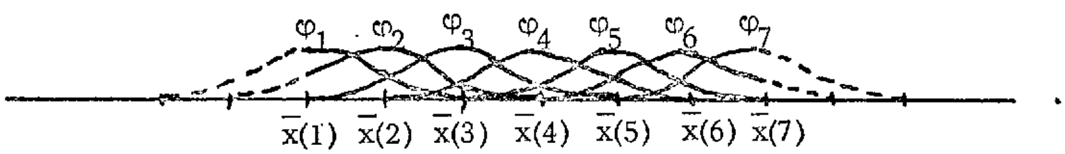
equation (24) is the matrix form of equation (30). Adding constraint (27) gives a ℓ_1 formulation to obtain $f_a(x)$ that satisfies equation (8) and (9).

8. Remarks

The B-spline basis functions used gives a sparce coefficient matrix A^T. The central B-splines are used even as the end splines with the support outside the interval $[\bar{x}(1), \bar{x}(m)]$. For equally spaced knots this presents no problem. For m = 7 the F matrix is

$$F = 1/6 \begin{bmatrix} 4 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 4 \end{bmatrix}$$

which corresponds to



This formulation has a diagonally dominant, symmetric matrix F, which has a nice closed form for F^{-1} given by

$$F^{-1} = \frac{6}{b_m} [a_{ij}] \text{ is the m x m matrix}$$

where

$$a_{ij} = \begin{cases} b_{i-1} b_{m-1} \quad , & \text{if} \quad i = j \\ (-1)^{i+j} b_{i-1} b_{m-1} \quad , & \text{if} \quad j > i \\ a_{ji} \quad , & \text{if} \quad j < i \end{cases}$$

where $b_0 = 1$, $b_1 = 4$, $b_k = 4b_{k-1} - b_{k-2}$, $k = 2, 3, \ldots, m$. See [19].

This of course makes solving equation (21) trivial. For more knots there will be a much larger percentage of zero coefficients as the size of the F matrix increases, hence, the A^T matrix will be sparce for large m. The revised simplex algorithm leaves the zero entries of the initial trableau zero at each iteration. This is not true for the simplex algorithm. The algorithm of Barrodale uses the simplex algorithm. Because of its speed perhaps, it could be adapted to include the area matching constraint (27) and still be very fast.

The ℓ_1 formulation of A is a (2m) x (2m) matrix so no computational speed can be expected by solving the dual of equations (24) and (27). Where as for the ℓ_∞ norm, A is a (m+1) x (2m) matrix and some computational improvement might occur from solving the dual system of equations (24) and (27)..

9. Algorithm for $f_a(x)$

(1) Choose $F(x) = \int_{-\infty}^{x} f(s)\,ds$

(2) Generate random numbers Z_i in $(0, 1)$, $i = 1, 2, \ldots, n$.

(3) For each i find $F^{-1}(Z_i) = x_i$, $i = 1, 2, \ldots, n$.

(4) Order the $\{x_i\}_{i=1}^{n}$ in increasing order.

(5) Choose knots $\{\bar{x}(i)\}_{i=1}^{m}$.

(6) Form a normalized histogram of the data with respect to the knots to obtain $\{\bar{x}(i), y_i)\}_{i=1}^{m}$.

(7) Use the revised simplex algorithm for an ℓ_1 or ℓ_∞ fit to $\{\bar{x}(i), y_i)\}_{i=1}^{m}$ with cubic B-splines as basis elements to obtain $\{a_i\}_{i=1}^{m}$ and hence $f_a(x)$.

(8) Compare $f(x)$ and $f_a(x)$.

For raw data use steps (4) to (7) in the algorithm.

10. Example(Bliss Histogram [3])

This example was used so as to compare what is obtained by this algorithm with the Histospline [4] and the Splinogram [14]. Let

$$\chi_{[a,b]} = \begin{cases} 1, & \text{if} \quad x \in [a,b] \\ 0, & \text{Otherwise,} \end{cases}$$

then the Histogram H is defined by

$$H = \sum_{i=11}^{24} h_i \chi_{[x_i,\, x_{i+1}]}$$

where

$h_1 = 1/578$	$h_8 = 104/578$
$h_2 = 5/578$	$h_9 = 66/578$
$h_3 = 20/578$	$h_{10} = 44/578$
$h_4 = 38/578$	$h_{11} = 18/578$
$h_5 = 50/578$	$h_{12} = 10/578$
$h_6 = 110/578$	$h_{13} = 1/578$
$h_7 = 110/578$	$h_{14} = 1/578$

and

$$\{x_i = i + 9.5\}_{i=1}^{15}$$

This data is normally distributed. The raw data was not given so the algorithm had to start from the histogram. $f_a(x)$ for this example is unimodal.

For the ℓ_∞ approximation to the data points, $\gamma = .04370$ the $\{a_i\}_{i=1}^{m}$ obtained were

a(1) = .00144	a(8) = .19959
a(2) = .00457	a(9) = .09850
a(3) = .03130	a(10) = .08739
a(4) = .07663	a(11) = 0.0
a(5) = .05337	a(12) = .01032
a(6) = .22587	a(13) = 0.0
a(7) = .17713	a(14) = 0.0

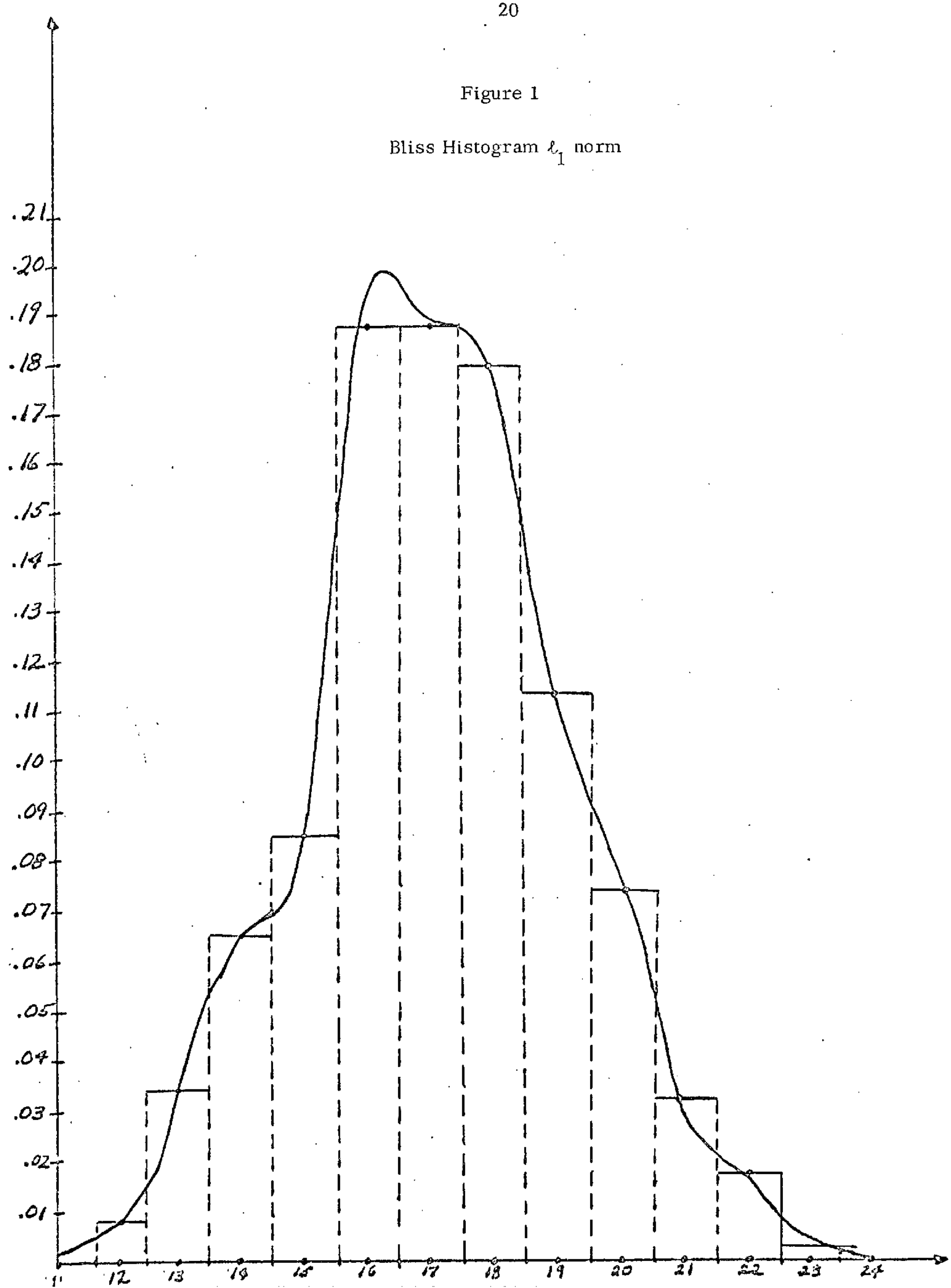
The graph of the function is in Figure 1.

For the ℓ_1 approximation to the data point $\gamma = .04223$

a(1) = .00144	a(8) = .20002
a(2) = .00457	a(9) = .09334
a(3) = .03130	a(10) = .09334
a(4) = .07663	a(11) = .01777
a(5) = .05337	a(12) = .02136
a(6) = .22590	a(13) = 0.0
a(7) = .17702	a(14) = .00258

Figure 1

Bliss Histogram ℓ_1 norm



11. Conclusion

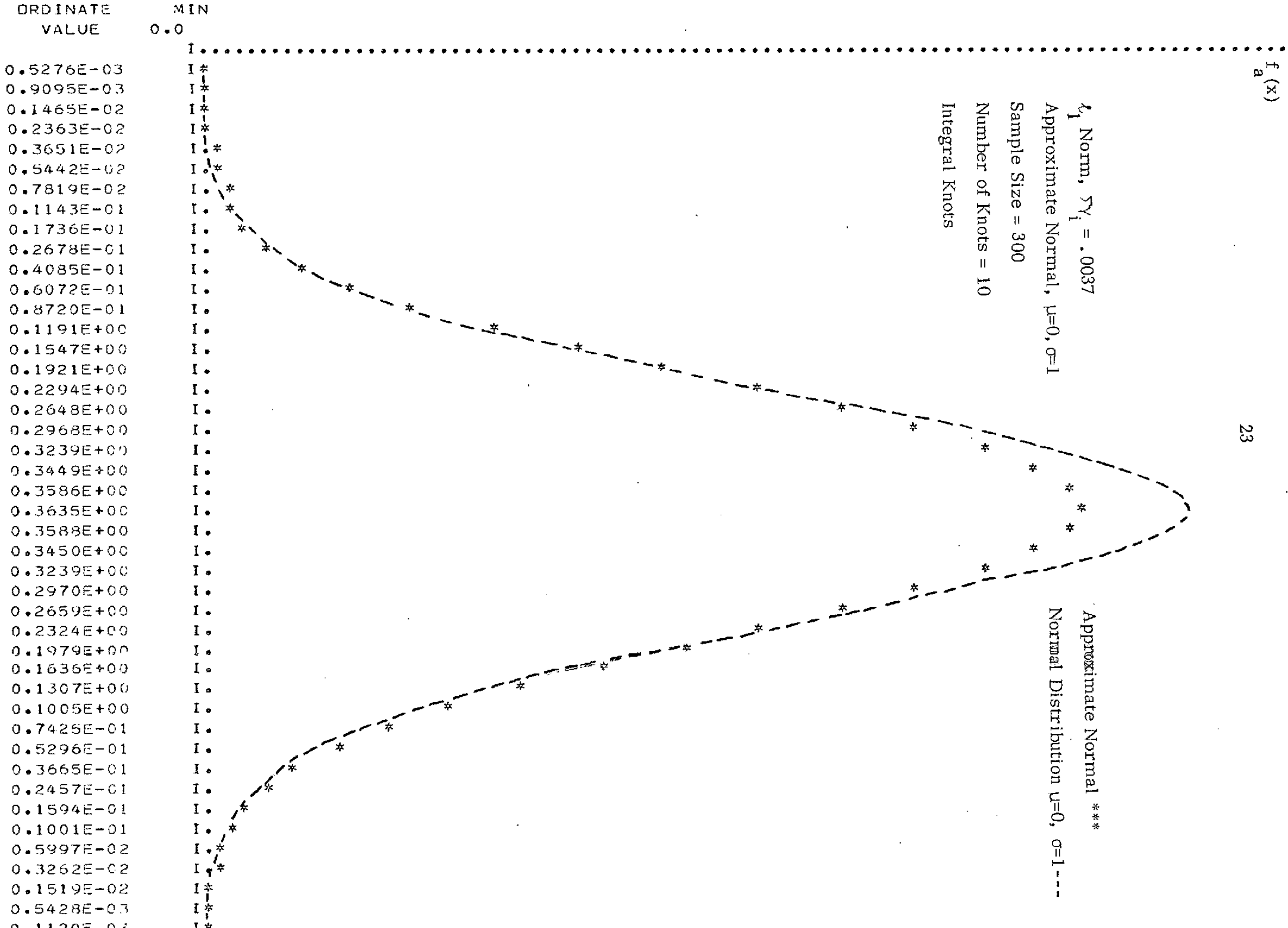
The Histospline of [4] can go negative and is not unimodal, where as $f_a(x)$ cannot go negative and was unimodal for the example of the Bliss histogram. The splinogram of [14] was called to my attention after this algorithm was completed but not written up. The splinogram for the Bliss histogram appears smoother than $f_a(x)$ in the ℓ_∞ or ℓ_1 norm, but $f_a(x)$ has one more continuous derivative than the splinogram. Also, $f_a(x)$ could by just changing k in the program have as many continuous derivatives as desired, however there would be a loss of the tridiagonal structure of F hence, computing the $\{a_i\}_{i=1}^m$ would take longer. The A^T matrix could be modified to where it had the area matching property at each point $\{\bar{x}(i)\}_{i=1}^m$ like the Histospline and still satisfy equation (8). So this formulation is more versitle than either of the previous ones. Also, it is not necessary to require a histogram in the algorithm and alternate methods that by pass this requirement could be substituted.

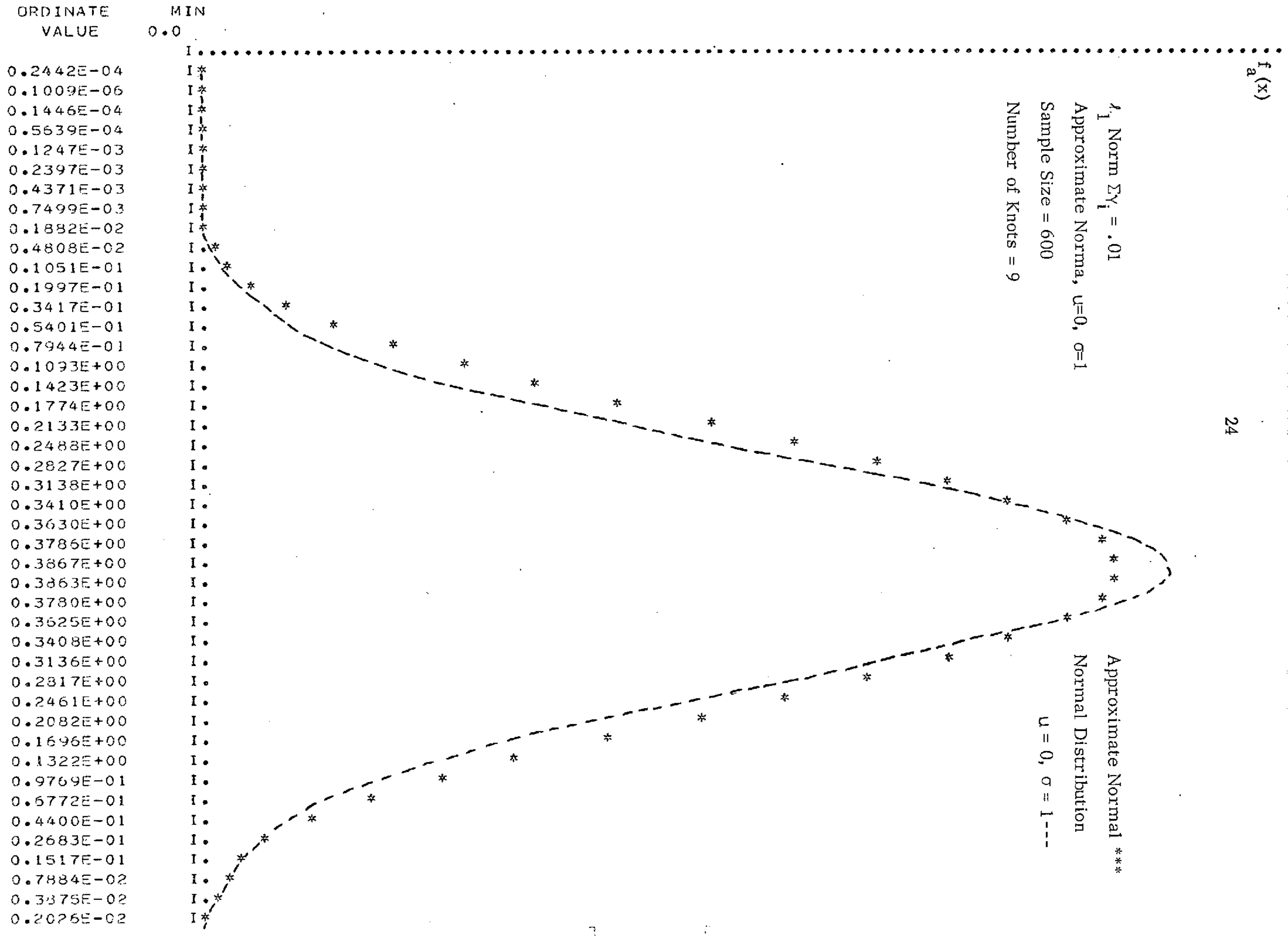
The algorithm should be tested for recovering several probability density functions.

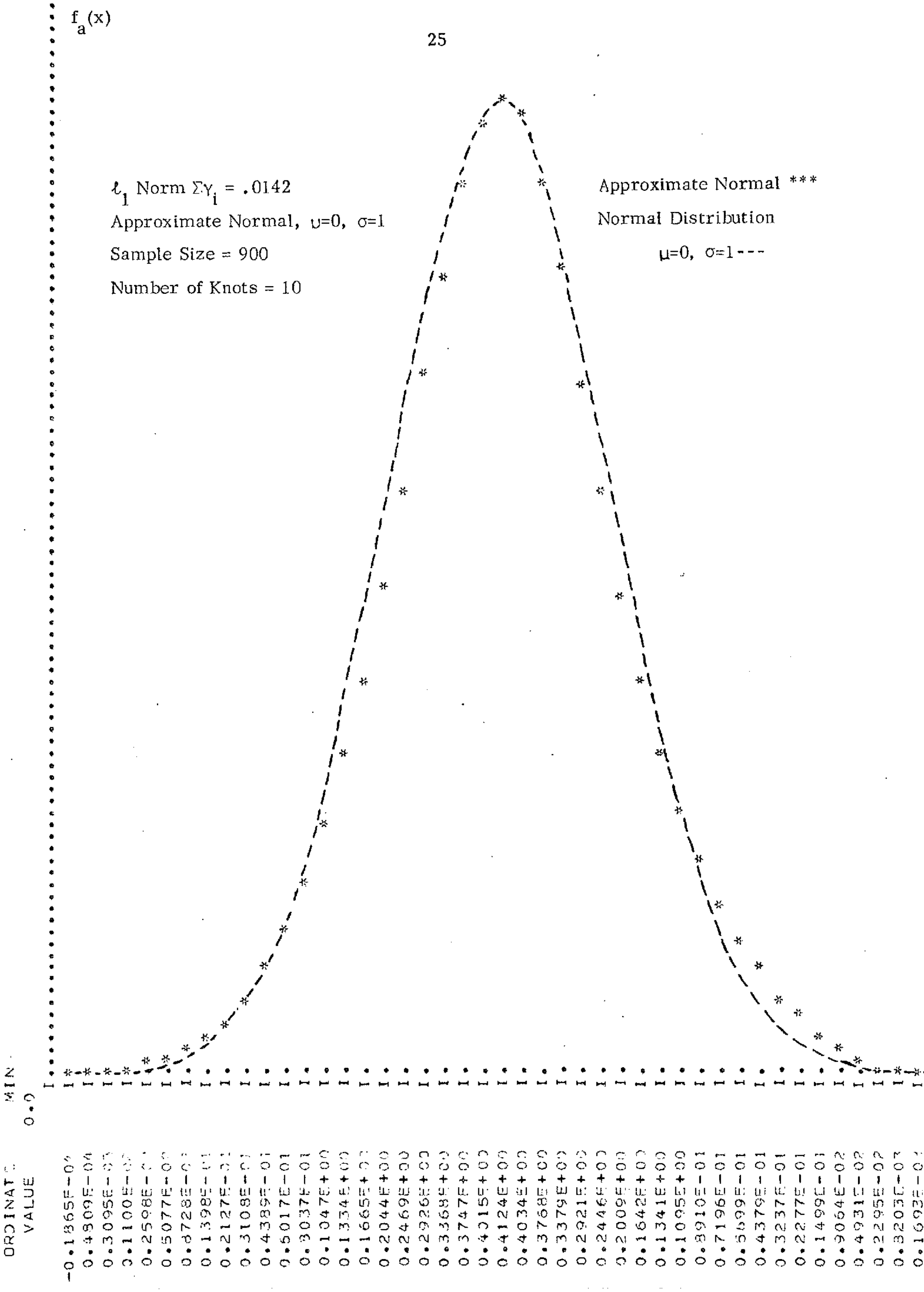
12. Appendix A. Test of Normal Distribution

To investigate the algorithm for the normal distribution, normal random samples of size 300, 600, and 900 were generated using the algorithm of Moshman [18]. The algorithm of this paper is then used to obtain the approximating function for these sample sizes. The graphs of the approximating functions are compared with the graph of the normal distribution for sample sizes 300, 600, and 900 on pages 23, 24, and 25, respectively.

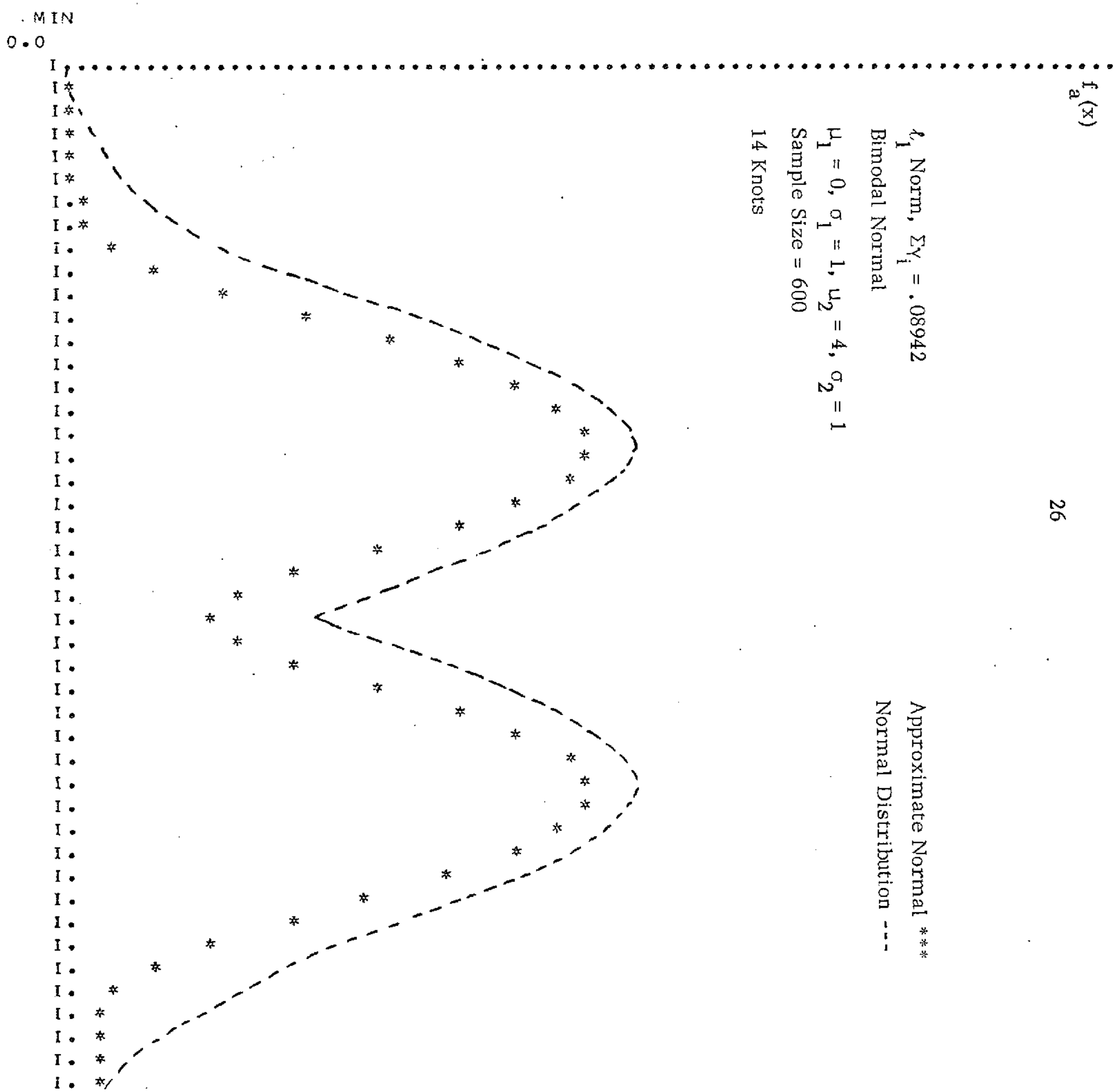
A bimodal normal distribution is also tested to see how well the algorithm distinguishes between the unimodal and bimodal functions. The results for this test is found on page 26.


ORDINATE VALUE
MIN 0.0
0.5276E-03
0.9095E-03
0.1465E-02
0.2363E-02
0.3651E-02
0.5442E-02
0.7819E-02
0.1143E-01
0.1736E-01
0.2678E-01
0.4085E-01
0.6072E-01
0.8720E-01
0.1191E+00
0.1547E+00
0.1921E+00
0.2294E+00
0.2648E+00
0.2968E+00
0.3239E+00
0.3449E+00
0.3586E+00
0.3635E+00
0.3588E+00
0.3450E+00
0.3239E+00
0.2970E+00
0.2659E+00
0.2324E+00
0.1979E+00
0.1636E+00
0.1307E+00
0.1005E+00
0.7425E-01
0.5296E-01
0.3665E-01
0.2457E-01
0.1594E-01
0.1001E-01
0.5997E-02
0.3262E-02
0.1519E-02
0.5428E-03
0.1120E-03
$f_a(x)$
ℓ_1 Norm, $\Sigma\gamma_i$ = .0037
Approximate Normal, μ=0, σ=1
Sample Size = 300
Number of Knots = 10
Integral Knots
Approximate Normal ***
Normal Distribution μ=0, σ=1 ---


ORDINATE VALUE
MIN 0.0
0.2442E-04
0.1009E-06
0.1446E-04
0.5639E-04
0.1247E-03
0.2397E-03
0.4371E-03
0.7499E-03
0.1882E-02
0.4808E-02
0.1051E-01
0.1997E-01
0.3417E-01
0.5401E-01
0.7944E-01
0.1093E+00
0.1423E+00
0.1774E+00
0.2133E+00
0.2488E+00
0.2827E+00
0.3138E+00
0.3410E+00
0.3630E+00
0.3786E+00
0.3867E+00
0.3863E+00
0.3780E+00
0.3625E+00
0.3408E+00
0.3136E+00
0.2817E+00
0.2461E+00
0.2082E+00
0.1696E+00
0.1322E+00
0.9769E-01
0.6772E-01
0.4400E-01
0.2683E-01
0.1517E-01
0.7884E-02
0.3875E-02
0.2026E-02
$f_a(x)$
ℓ_1 Norm $\Sigma Y_i = .01$
Approximate Norma, u=0, σ=1
Sample Size = 600
Number of Knots = 9
Approximate Normal ***
Normal Distribution
u = 0, σ = 1---


$f_a(x)$
ℓ_1 Norm $\Sigma\gamma_i$ = .0142
Approximate Normal, μ=0, σ=1
Sample Size = 900
Number of Knots = 10
Approximate Normal ***
Normal Distribution
μ=0, σ=1 ---
MIN
0.0
ORDINATE
VALUE
-0.1865E-04
0.4809E-04
0.3095E-03
0.1100E-02
0.2598E-02
0.5077E-02
0.8728E-02
0.1398E-01
0.2127E-01
0.3108E-01
0.4389E-01
0.6017E-01
0.8037E-01
0.1047E+00
0.1334E+00
0.1665E+00
0.2044E+00
0.2469E+00
0.2926E+00
0.3368E+00
0.3747E+00
0.4015E+00
0.4124E+00
0.4034E+00
0.3768E+00
0.3379E+00
0.2921E+00
0.2446E+00
0.2009E+00
0.1642E+00
0.1341E+00
0.1095E+00
0.8910E-01
0.7196E-01
0.5699E-01
0.4379E-01
0.3237E-01
0.2277E-01
0.1499E-01
0.9064E-02
0.4931E-02
0.2295E-02
0.3203E-03

ITEM COUNT	ORDINATE VALUE
1	0.3123E-03
2	0.6337E-03
3	0.1019E-02
4	0.1397E-02
5	0.1713E-02
6	0.2521E-02
7	0.6022E-02
8	0.1470E-01
9	0.3093E-01
10	0.5508E-01
11	0.8376E-01
12	0.1133E+00
13	0.1402E+00
14	0.1619E+00
15	0.1774E+00
16	0.1856E+00
17	0.1857E+00
18	0.1775E+00
19	0.1619E+00
20	0.1394E+00
21	0.1113E+00
22	0.8248E-01
23	0.5980E-01
24	0.5007E-01
25	0.5831E-01
26	0.8045E-01
27	0.1096E+00
28	0.1388E+00
29	0.1623E+00
30	0.1786E+00
31	0.1866E+00
32	0.1856E+00
33	0.1754E+00
34	0.1577E+00
35	0.1344E+00
36	0.1074E+00
37	0.7907E-01
38	0.5239E-01
39	0.3048E-01
40	0.1626E-01
41	0.9665E-02
42	0.7949E-02
43	0.8267E-02
44	0.7965E-02



References

1. BARRODALE, I., and ROBERTS, F.D.K., An Improved Algorithm for Discrete ℓ_1 Linear Approximation, University of Wisconsin, Madison, Wisconsin, Mathematical Research Center, TR-1172, 1972.

2. BERGER, S.A., WEBSTER, W.C., TAPIA, R.A., and ATKINS, D.A., Mathematical Ship Lotting, Journal of Ship Research, Vol. 10, No. 4, 1966.

3. BLISS, C.I., Statistics in Biology, Vol. I, McGraw-Hill, 1967.

4. BONEVA, L.I., KENDALL, D., and STETANOV, I., Spline Transformations: Three New Diagnostic Aids for the Statistical Data-Analyst, Royal Statistical Society meeting Research Section, October 14, 1970, M.J.R. Healy, Chairman.

5. CHARNES, A., COOPER, W.W., and FERGUSON, R.O., Optimal Estimation of Executive Compensation by Linear Programming, Management Science, 1955.

6. CURRY, H.B., and SCHOENBERG, I.J., On Polya Frequency Functions IV The Fundamental Spline Functions and Their Limits, Journal Analyses Math Vol. 17, 1966.

7. DANTZIG, G.B., Linear Programming and Extensions, Princeton University Press, 1963.

8. FOURIER, J.B., Oeuvres De Fourier,edited by George Darboux, Paris, Gauthier-Villars, 1888-1890, Vol. 2.

9. GREVILLE, T.N.E., The General Theory of Osculatory Interpolation, Transactions of the Actuarial Society of America, Vol. 45, 1944.

10. GREVILLE, T.N.E., Theory and Applications of Spline Functions, Academic Press, New York, 1968.

11. HOLLADAY, J.C., Smoothest Curve Approximation, Math Tables Aids to Computation, Vol. 11, 1957.

12. ROSEN, J.B., An Interactive Display for Approximation by Linear Programming, Communication of the Association for Computer Machinery, Vol. 13, No. 11, 1970.

13. SCHOENBERG, I.J., Contributions to the Problem of Approximation of Equidistant Data by Analytic Functions, Quarterly Applied Mathematics, Vol. 4, 1946.

14. SCHOENBERG, I.J., Notes on Spline Functions II, On the Smoothing of Histograms, University of Wisconsin, Mathematical Research Center, TR-1222, 1970.

15. SCHUMAKER, L.L., Some Algorithms for the Computation of Interpolating and Approximating Spline Functions, Theory and Applications of Spline Functions, Edited by T.N.E. Griville, 1968.

16. SPIVEY, A., and THRALL, R.M., Linear Optimization, Holt, Rinehart, and Winston, New York, 1971

17. WAGNER, H.M., Linear Programming Techniques for Regression Analysis, Journal of American Statistical Association, Vol. 54, 1959.

Additional Bibliography

18. MOSHMAN, J., Random Number Generators, Edited by A. Ralston and H.S. Wilf, Mathematical Methods for Digital Computers, Vol. II, John Wiley and Son, New York, 1960.

19. GREGORY, R.T., and KARNEY, D.L., A Collection of Matrices for Testing Computational Algorithms, John Wiley and Sons, New York, 1971.